



# UNITED STATES PATENT AND TRADEMARK OFFICE

11/11  
UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/057,135	10/29/2001	John Joseph Mazzitelli	100110992-1	1932
7590	03/10/2005		EXAMINER	
HEWLETT-PACKARD COMPANY Intellectual Property Administration P.O. Box 272400 Fort Collins, CO 80527-2400			SERRAO, RANODHI N	
			ART UNIT	PAPER NUMBER
			2141	

DATE MAILED: 03/10/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/057,135	MAZZITELLI, JOHN JOSEPH	
	Examiner Ranodhi Serrao	Art Unit 2141	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

### **Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
  - If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
  - If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
  - Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1)  Responsive to communication(s) filed on 29 October 2001.

2a)  This action is **FINAL**.                            2b)  This action is non-final.

3)  Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4)  Claim(s) 1-30 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5)  Claim(s) \_\_\_\_\_ is/are allowed.

6)  Claim(s) 1-30 is/are rejected.

7)  Claim(s) \_\_\_\_\_ is/are objected to.

8)  Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9)  The specification is objected to by the Examiner.

10)  The drawing(s) filed on 29 October 2001 is/are: a)  accepted or b)  objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)  The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12)  Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a)  All b)  Some \* c)  None of:  
1.  Certified copies of the priority documents have been received.  
2.  Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3.  Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1)  Notice of References Cited (PTO-892)  
2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3)  Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_  
4)  Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.  
5)  Notice of Informal Patent Application (PTO-152)  
6)  Other: \_\_\_\_\_.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-30 are rejected under 35 U.S.C. 102(e) as being anticipated by LiVecchi (6,427,161).

As per claim 1, LiVecchi teaches a multi-threaded server accept method (column 6, line 61-column 7, line 7); creating a socket accept thread by a control thread of a server process (column 11, line 66-column 12, line 21); receiving a service request from a client by the socket accept thread (column 2, line 62-column 3, line 6); transferring the request to a data structure; and retrieving the request, by the control thread, from the data structure (column 11, lines 1-37: wherein linked lists function as data structures); transferring the request to a client thread, by the control thread, to process request data associated with the request, and processing the request data by the client thread (column 7, line 16-column 8, line 37).

As per claims 2, 13, and 23, LiVecchi teaches the data structure comprises a queue (column 11, lines 1-37).

As per claims 3, 14, and 24, LiVecchi teaches the data structure comprises a FIFO queue (column 11, lines 1-37).

As per claim 4, LiVecchi teaches waiting for service requests by performing an accept () call (column 11, lines 1-37).

As per claim 5, LiVecchi teaches receiving the request comprises receiving a client socket object (column 6, lines 13-30).

As per claim 6, LiVecchi teaches waiting for the service request from the client by the socket accept thread (column 3, lines 51-67).

As per claim 7, LiVecchi teaches creating the client thread to process the request data (column 6, lines 31-48).

As per claim 8, LiVecchi teaches receiving a second request by the socket accept thread from the client (column 4, lines 10-21); transferring the second request to the data structure (column 11, lines 1-37); retrieving the second request by the control thread (column 15, lines 15-36); transferring the second request to a second client thread to process second request data; and processing the second request data by the second client thread (column 7, line 16-column 8, line 37).

As per claim 9, LiVecchi teaches creating the second client thread to process the second request data (column 11, lines 1-37).

As per claim 10, LiVecchi teaches socket accept thread and the control thread are executed on a single processor (column 1, lines 19-40).

As per claim 11, LiVecchi teaches the steps of transferring the request to the data structure and retrieving the request from the data structure are serially performed (column 12, lines 17-21: wherein pending connections on the queue is being performed serially).

As per claim 12, LiVecchi teaches a multi-thread server accept system, comprising: a server process residing on a server (column 6, line 61-column 7, line 7); and operable to create a socket accept thread by a control thread of a server process residing on the server (column 11, line 66-column 12, line 21); receive a service request from a client by the socket accept thread (column 2, line 62-column 3, line 6); transfer the request to a data structure; retrieve the request, by the control thread, from the data structure (column 11, lines 1-37: wherein linked lists function as data structures); transfer the request to a client thread, by the control thread, to process request data associated with the request; and process the request data by the client thread (column 7, line 16-column 8, line 37).

As per claim 15, LiVecchi teaches the socket accept thread is operable to wait for service requests by performing an accept() call (column 11, lines 1-37).

As per claim 16, LiVecchi teaches the socket accept thread is operable to receive the request by receiving a client socket object from the client (column 6, lines 13-30).

As per claim 17, LiVecchi teaches the server process is further operable to create the client thread to process the request data (column 6, lines 31-48).

As per claim 18, LiVecchi teaches the server process is further operable to: receive a second request from the client by socket accept thread after transferring the request to the data structure (column 4, lines 10-21); transfer the second request to the data structure (column 11, lines 1-37); retrieve the second request by the control thread (column 15, lines 13-36); transfer the second request to a second client thread to

process the second request data; and process the second request data by the second client thread (column 7, line 16-column 8, line 37).

As per claim 19, LiVecchi teaches the server process is further operable to create the second client thread to process the second request data (column 11, lines 1-37).

As per claim 20, LiVecchi teaches the socket accept thread and the control thread are executed on a single processor (column 1, lines 19-40).

As per claim 21, LiVecchi teaches the server process is further operable to serially perform the steps of transferring the request to the data structure and retrieving the request from the data structure (column 12, lines 17-21: wherein pending connections on the queue is being performed serially).

As per claim 22, LiVecchi teaches a multi-threaded server accept application, comprising an application software residing on a computer-readable medium (column 6, line 61-column 7, line 7) and operable to: create a socket accept thread by a control thread of the application software (column 11, line 66-column 12, line 21); receive a request from a client by the socket accept thread (column 2, line 62-column 3, line 6); transfer the request to a data structure; retrieve the request, by the control thread, from the data structure (column 11, lines 1-37: wherein linked lists function as data structures); transfer the request to a client thread, by the control thread, to process request data associated with the request; and process the request data by the client thread (column 7, line 16-column 8, line 37).

As per claim 25, LiVecchi teaches the application software is further operable to wait for service requests by calling an accept() program (column 11, lines 1-37).

As per claim 26, LiVecchi teaches the application is further operable to receive the request by receiving a client socket object from the client (column 6, lines 13-30).

As per claim 27, LiVecchi teaches the application software is further operable to create the client thread to process the request data (column 6, lines 51-67).

As per claim 28, LiVecchi teaches the application software is further operable to: receive a second request from the client by the socket accept thread after transferring the request to the data structure (column 4, lines 10-21); transfer the second request to the data structure (column 11, lines 1-37); retrieve the second request by the control thread (column 15, lines 13-36); transfer the second request to a second client thread to process second request data; and process the second request data by the second client thread (column 7, line 16-column 8, line 37).

As per claim 29, LiVecchi teaches the socket accept thread and the control thread are executed on a single processor (column 1, lines 19-40).

As per claim 30, LiVecchi teaches the application software is further operable to serially perform the steps of transferring the request to the data structure and retrieving the request from the data structure (column 12, lines 17-21: wherein pending connections on the queue is being performed serially).

***Conclusion***

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Hanif et al. (5,796,954) teaches a method and system for maximizing the use of threads in a file server for processing network requests. Woods et al. (6,594,590) teaches a method and apparatus for improved inspection and classification of attributes of a workpiece. Bayeh et al. (6,633,914) teaches systems, methods and computer program products for handling client requests for server application processing using a thread pool. Stanbach (2001/0049747) teaches a multi-threaded name server.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ranodhi Serrao whose telephone number is (571)272-7967. The examiner can normally be reached on 8:00-5:30pm, M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Rupal Dharia can be reached on (571)272-3880. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



RUPAL DHARIA  
SUPPLYING PATENT EXAMINER